DOI 10.1007/s00165-014-0304-0
BCS © 2014
Formal Aspects of Computing (2015) 27: 79–102

Formal Aspects of Computing

# Formal probabilistic analysis of detection properties in wireless sensor networks

Maissa Elleuch[1,2], Osman Hasan[2], Sofiène Tahar[2] and Mohamed Abid[1]

[1] CES Laboratory, National School of Engineers of Sfax, Sfax University, Soukra Street, 3052 Sfax, Tunisia
[2] Department of Electrical and Computer Engineering, Concordia University, 1455 de Maisonneuve W., Montreal, QC H3G 1M8, Canada

**Abstract.** In the context of wireless sensor networks (WSNs), the ability to detect an intrusion event is the most desired characteristic. Due to the randomness in nodes scheduling algorithm and sensor deployment, probabilistic techniques are used to analyze the detection properties of WSNs. However traditional probabilistic analysis techniques, such as simulation and model checking, do not ensure accurate results, which is a severe limitation considering the mission-critical nature of most of the WSNs. In this paper, we overcome these limitations by using higher-order-logic theorem proving to formally analyze the detection properties of randomly-deployed WSNs using the randomized scheduling of nodes. Based on the probability theory, available in the HOL theorem prover, we first formally reason about the intrusion period of any occurring event. This characteristic is then built upon to develop the fundamental formalizations of the key detection metrics: the detection probability and the detection delay. For illustration purposes, we formally analyze the detection performance of a WSN deployed for border security monitoring.

**Keywords:** Theorem proving, Wireless sensor networks, Scheduling, Performance analysis, Detection probability, Detection delay

## 1. Introduction

Wireless sensor networks (WSNs) [YMG08] guarantee a continuous and automated monitoring of a given field without any human presence. This distinguishing feature is attained trough deploying a collection of battery-powered and wirelessly-connected miniature devices over the area of interest. The main task of such devices is to take measurements of the surrounding environment, to a base station to perform a centralized decision mechanism. Nowadays, wireless sensor networks are extensively being deployed in a wide range of real-world applications, such as home automation, detection of natural disasters, biological attacks and military tracking.

*Correspondence and offprint requests to*: M. Elleuch. E-mail: melleuch@ece.concordia.ca

Since nodes are usually stand-alone and battery powered, extending the network lifetime is very critical [YMG08]. Therefore, the k-set randomized nodes scheduling [JS07, LC08, LWXS06, XCW+10] is commonly applied to preserve energy. The main idea of such approach is to randomly organize the nodes into alternatively working sub-networks. Hence, during a given time slot, only the nodes belonging to the current active sub-network are powered up and may report an occurring event while all the other nodes are inactive and thus contribute to the power saving of the overall system.

In general, a wireless sensor network is expected to always report occurring events at any point of the monitored area to a base station with a short delay. This feature determines the detection abilities of the whole network and is measured through two key performance attributes: the detection probability and the detection delay. More specifically, the detection probability is the probability of detecting an occurring event within the monitored area [XCW+10]. Due to the randomness in the nodes scheduling approach coupled with the unpredictable deployment of sensors, the detection characteristic cannot be usually ensured. Indeed, there is a possibility that an occurring event may not be detected if there are no nodes deployed in its surrounding area or the deployed nodes are inactive, due to random scheduling. Such situations will also lead to an infinite detection delay which is not desired at all. However, in most WSN applications, the network has to react according to intrusions detection. For example, in a WSN deployed for forest fire detection, the outbreak of a fire should be simultaneously reported with the highest probability and the minimum delay, in order to alert the user. Consequently, missing an intrusion event can be really disastrous in the context of mission-critical WSN applications. Thus, probabilistic techniques are used to judge the detection properties of WSNs with the goal to maximize the probability of detection and minimize the detection delay.

Traditionally, paper-and-pencil proof based probabilistic techniques have been used to analyze the performance of random scheduling for WSNs [LWXS06, LC08]. Simulation, using the Monte Carlo method [Mac98], is then used to validate the analytical results, which can be error-prone. Due to the inherent incompleteness of simulation coupled with the rounding errors of computer arithmetics, such results cannot be considered as 100 % accurate, which is a serious limitation for mission-critical WSNs.

Formal methods [Abr09] can overcome the limitations of simulation and have been used to validate a wide range of hardware and software systems. Such methods enhance the analysis reliability using rigorous mathematical techniques to model and verify the given system. Formal methods have also been explored for analyzing WSNs but most of the existing work is focused on analyzing their functional aspects only. However, given the wide application of WSNs in safety and mission-critical domains, there is a dire need to accurately assess their performance as well. With this motivation, this paper provides a formal approach for an accurate performance analysis of the probabilistic detection properties of WSNs using the k-set randomized scheduling.

We primarily build upon the recently developed probability theory available in the HOL theorem prover [Mha12], to formally analyze the detection properties of the k-set randomized scheduling algorithm. The choice of using higher-order-logic allows us to model any system including its random and unpredictable components [Mha12]. In [EHTA11], we presented the HOL formalization of the coverage property in WSNs. The efficiency of our higher-order-logic developments have been shown on a real-world WSN application for forest fire detection [EHTA13]. In this paper, we provide a development regarding detection properties in WSN, which include the detection probability and the detection delay. The practical effectiveness of the developed formalizations is illustrated through formally analyzing the asymptotic detection behavior of a real-world WSN for border surveillance. Thanks to the proposed approach, this is the first time, to the best of our knowledge, that the performance analysis of this kind of a WSN application is analyzed in a complete formal manner.

The rest of this paper is organized as follows: Sect. 2 reviews related work on the validation of WSN algorithms. We briefly present, in Sect. 3, the main HOL requirements that we build upon in this work. The k-set randomized scheduling algorithm is introduced in this section as well. In Sect. 4, we describe our higher-order-logic formalizations of the key detection properties: the detection probability and the detection delay. The practical effectiveness of these formal results is illustrated, in Sect. 5, through a WSN application for border monitoring security. Section 6 is devoted to discuss the main results of our work. Finally, Sect. 7 concludes the paper.

## 2. Related work

Due to its wide applicability, the random scheduling algorithm has been analyzed using various approaches in the open literature. Paper-and-pencil analysis is indeed the most commonly used approach for the performance analysis of this algorithm. In such analysis, a mathematical model is built by first identifying the required random variables and the corresponding performance attributes. Then, a rigorous analysis based on the theoretical foundations of probability is done. In order to validate the theoretical analysis, simulations are done using the Monte Carlo method [Mac98]. In [LWXS06, LC08], a coverage-based random scheduling is analyzed using a mathematical model. Evaluations are done using a Java simulator by setting the monitored region to 200 m×200 m, the detection range to 10 m, and the subsets to 6. In [XZP+09], theoretical analysis is conducted to validate the coverage performance of randomized scheduling in the context of an hybrid surveillance framework for environmental monitoring. Results are validated through simulation on a circular surface of a radius $R = 10000$, where up to $n = 2000$ nodes are uniformly deployed. Due to the inherent nature of simulation coupled with the usage of computer arithmetic, these probabilistic analysis results cannot be termed as 100 % accurate. Moreover, the analysis results are not generic, i.e., they are specific to a region, range and number of subsets.

Most of the existing literature in the formal analysis of WSNs utilizes traditional model checking [CGP00] to validate many aspects of WSNs. In [OT07], the authors performed the formal analysis of the Optimal Geographical Density Control algorithm (OGDC) in the RT-Maude rewriting tool [RTM] by verifying the network coverage intensity and lifetime. In [HRZ08], model checking is used to verify WSNs security aspects in the SLEDE framework. Similarly, a model checking based framework, called NesC@PAT [ZSL+11], is also used for verifying WSNs implementations in NesC.

In addition to its accuracy, the main advantage of model checking is its mechanization. However, model checking also suffers from some major shortcomings, like the common problem of state-space explosion [CGP00], where the size of the state-based model increases exponentially as the complexity of the given WSN grows. Such problems have been noticed in most of the works [HRZ08, ZSL+11]. For example, in [ZSL+11], it is reported that over 1 million states are generated in order to verify a single property. Moreover, in [HRZ08], additional temporal abstractions and some parameters reduction have been applied to enhance the scalability of the analysis. Finally, the mentioned works do not allow capturing randomness of WSNs into account, which is a strict limitation since most of the WSN algorithms are probabilistic. The authors of [OT07] have besides suggested the use of PMaude [AMS06] to enhance their probabilistic analysis.

Probabilistic model checking [RKNP04] has also been successfully used for the probabilistic functional analysis of wireless systems. Probabilistic model checking has the same principles as traditional model checking: the mathematical model of the probabilistic system is exhaustively tested to check if it meets a set of probabilistic properties. The probabilistic model checker PRISM [PRI] has been used quite frequently for the verification of Medium Access Control (MAC) protocols for WSNs [FHM07, Fru06, ZBA10]. Nevertheless, the accuracy of probabilistic model checking is very limited when reasoning about statistical properties. For example, in [ZBA10], the expected values of latency and energy have been verified by running several experiments. The obtained results were hence specific to the chosen configurations and can never be considered as generic.

Besides model checking, higher-order-logic theorem proving [GM93] has also been used for analyzing WSN algorithms. In [BMP09], a WSN algorithm is formally modelled, within the PVS system, by utilizing a library of mathematically specified sub-blocks, like the nodes, the network structure, communication primitives and protocols. Furthermore, the resulted framework is enriched by some theories expressing probabilistic scenarios like nodes mobility and link quality changes. The feasibility of this framework is illustrated by manually analyzing the trace execution of the Surge algorithm [BMP08], and formally verifying the correctness of the message delivery for the reverse path forwarding algorithm [BMP09]. Nevertheless, the randomness here is modeled by using a pseudo-random generator, which compromises the accuracy of the analysis results.

While most of the previous works on the formal analysis of WSNs have clearly recognized their inherent modelling limits regarding the probabilistic feature, we used the probabilistic analysis foundations available in the HOL theorem prover to formally verify the coverage performance properties of the k-set randomized algorithm in [EHTA11]. The results have been found to be absolutely accurate since a measure theoretic probability theory is used to analyze the WSN algorithm within the sound core of a theorem prover. In [EHTA13], we demonstrated the practical effectiveness of these results on a real-world WSN application for forest fire detection. Nevertheless, the network coverage reflects the detection characteristics of the network only in the case of long events. In the current paper, we are interested in formally analyzing the detection characteristics of wireless sensor networks using the k-set randomized scheduling for any kind of events.

**Table 1.** HOL symbols

| HOL symbol | Standard symbol | Meaning |
|---|---|---|
| $\wedge$ | $and$ | Logical $and$ |
| SUC $n$ | $n + 1$ | Successor of $n$ |
| count $n$ | $\{m \mid m < n\}$ | Set of all $m$ strictly less than $n$ |
| PREIMAGE $f$ $s$ | $\{x \mid f\ x \in s\}$ | The inverse image of the subset $s$ |
| $\{x \mid P(x)\}$ | $\{\lambda\, x.P(x)\}$ | The set of all $x$ that satisfy the condition $P$ |

## 3. Preliminaries

In this section, we first give an overview of the HOL theorem prover and the main required notations. Then, we briefly present the probability theory to conduct the probabilistic analysis in the HOL theorem prover. Some probability formalization results that we developed in HOL, are also provided in this section. The description of the k-set randomized scheduling algorithm is finally introduced to facilitate the understanding of the rest of the paper.

### 3.1. HOL theorem prover

The HOL theorem prover [HOL] is a proof assistant of higher-order logic. The verification approach of HOL is composed of three main steps: describing the system to be verified in higher-order logic, formalizing the properties of interest as proof goals of higher-order-logic and finally verifying these goals as theorems within HOL. Furthermore, the HOL theorem prover includes a very rich library of theories. A theory can be defined as a set of pre-verified theorems for a given domain, function or operation. When needed, a HOL theory can be loaded and used, which greatly aids the verification process. Additionally, users may be assisted by automatic proof procedures [GM93], which are a collection of steps in a single command. Despite the existence of all these theories and automatic procedures, most of the time, proofs in HOL are interactive and require the intervention of user. Various proof techniques, such as rewriting, simplification, specialization, generalization and mathematical induction, are available in HOL to aid the verification process. Table 1 summarizes some of the HOL symbols used in this paper and their corresponding mathematical interpretation [GM93].

### 3.2. Probabilistic analysis in HOL

Several works on the higher-order-logic formalization of probability theory exist in the open literature, (See e.g. [Hur02, Les07, Has08, APM09, HH11, Mha12]). In this work, we utilize the recently developed and most generic probability theory developed by Mhamdi [Mha12], within the HOL theorem prover. Unlike [HT08, HT07, HAA+09], such formalization has the merit of generalizing the previous HOL formalization of measure theory by including a Borel space [Bog06]. Through defining the extended real numbers in HOL, he formalized measure, Lebesgue, probability and information theories. Thus, the formalization of probability theory in HOL is based on the Kolmogorov axiomatic definition of probability. Such formalization has distinctly the advantage to provide a unified framework for discrete and continuous probability measures. In what follows, we give an overview of the foundational formalizations of the HOL probability theory.

A probability measure $P$ is basically a measure function on the sample space $\Omega$ and an event is a measurable set within the set $F$ of events, which are subsets of $\Omega$. Thus, $(\Omega, F, P)$ is a probability space iff it is a measure space and $P(\Omega) = 1$. A real random variable is specified in HOL in the following definition using the HOL function `real_random_variable`.

**Definition 3.1**

```
⊢ ∀X p. real_random_variable X p = prob_space p ∧
                                   ∀x ∈ p_space p ⇒ X x ≠ NegInf ∧ X x ≠ PosInf) ∧
                                   X ∈ measurable (p_space p,events p) Borel.
```

where $X$ designates the random variable which is by definition a real-valued measurable function (`measurable`) on a given probability space $p$ (`prob_space`) [MHT10]. A measurable function satisfies the condition that the inverse image of a measurable set is also measurable. The functions (`p_space p`) and (`events p`) are the definitions of the sample space $\Omega$ and the set of events $F$, respectively. The HOL symbols *NegInf* and *PosInf*, used in the above definition, are the higher-order-logic formalizations of negative infinity or positive infinity, whereas *Borel* is the HOL definition of the Borel sigma algebra. Mathematically, a Borel sigma algebra on a given space $A$ is the smallest sigma algebra generated by the open sets of $A$ [Bog06].

The probability distribution of a random variable is specified as the function that accepts a random variable $X$ and a set $s$ and returns the probability of the event $\{X \in s\}$. It has been formalized in HOL [MHT11] in Definition 3.2.

**Definition 3.2**

```
⊢ ∀X p. distribution p X = (λs. prob p (PREIMAGE X s ∩ p_space p)).
```

The expectation of a random variable $X$ is defined in HOL as its Lebesgue integral with respect to the probability measure $p$ [MHT11].

$$E[X] = \int_{\Omega} X dp. \tag{1}$$

The conditional probability has been also formalized in HOL [Liu13] according to the following mathematical definition.

$$Pr(A \mid B) = \frac{Pr(A \cap B)}{Pr(B)}. \tag{2}$$

where $A$ and $B$ are two events of the set $F$ of events.
Accordingly, the following useful results have been formally verified in HOL [Liu13].

- If the events $A$ and $B$ are independent such that $(Pr(B) \neq 0)$ , then

$$Pr(A \mid B) = Pr(A). \tag{3}$$

- The conditional probability of the event $(A \cup B)$, given the event $C$ is

$$Pr(A \cup B \mid C) = Pr(A \mid C) + Pr(B \mid C) - Pr(A \cap B \mid C). \tag{4}$$

- If $A$ and $B$ are disjoint, then the above equation becomes

$$Pr(A \cup B \mid C) = Pr(A \mid C) + Pr(B \mid C). \tag{5}$$

- The conditional probability of the event $(A \cap B)$ given the event $C$ is

$$Pr(A \cap B \mid C) = Pr(A \mid B \cap C) \times Pr(B \mid C). \tag{6}$$

- Given that $\{B_i, i \in s\}$, is a finite partition of the entire sample space $\Omega$, the law of total probability states that

$$Pr(A) = \sum_{i \in s} Pr(A \mid B_i) \times Pr(B_i). \tag{7}$$

The above equation has been formalized in HOL as follows.

**Theorem 3.1**

```
⊢ ∀p B A s. (prob_space p) ∧ FINITE s ∧ (A ∈ events p) ∧
  (∀x. x ∈ s ⇒ B x ∈ events p) ∧
  (∀a b. a ∈ s ∧ b ∈ s ∧ (a ≠ b) ⇒ DISJOINT (B a) (B b)) ∧
  (BIGUNION (IMAGE B s) = p_space p)
  ⇒ (prob p A = ∑_s (λi. (prob p (B i)) × (cond_prob p A (B i)))).
```

where

- The assumption (`∀x. x ∈ s ⇒ B x ∈ events p`) specifies a finite partition (`FINITE s`) of the whole outcome space Ω, i.e., a collection of events, which is pairwise disjoint (`∀a b. a ∈ s ∧ b ∈ s ∧ (a ≠ b) ⇒ DISJOINT (B a) (B b)`), and whose union is Ω (`BIGUNION (IMAGE B s) = p_space p`).
- `cond_prob` is the HOL formalization of the conditional probability.

Based on the above probability formalizations, we develop in HOL further mathematical notions required for the work described in this paper.

- Conditional independence: Two events $A$ and $B$ are conditionally independent given the event $C$, iff:

$$Pr(A \cap B \mid C) = Pr(A \mid C) \times Pr(B \mid C). \tag{8}$$

- The conditional independence is also equivalent to

$$Pr(A \mid B \cap C) = Pr(A \mid C). \tag{9}$$

- Discrete conditional expectation: The conditional expectation of the discrete random variable $X$ given the event $(Y = y)$, denoted by $E(X \mid Y = y)$, is the expected value of $X$ with respect to its conditional probability distribution, and is mathematically specified as follows

$$E(X \mid Y = y) = \sum_{x} x \times Pr(X = x \mid Y = y). \tag{10}$$

The concept of conditional expectation can be also extended to multiple events. In the current work, we will basically require the conditional expectation of $X$ given two events, i.e., $E(X \mid Y = y, Z = z)$, which is mathematically defined as

$$E(X \mid Y = y, Z = z) = \sum_{x} x \times Pr(X = x \mid Y = y \cap Z = z). \tag{11}$$

where $Z$ is a discrete random variable. Definition 3.3 gives the higher-order-logic formalization of the conditional expectation $E(X \mid Y = y, Z = z)$.

**Definition 3.3**

```
⊢ ∀X Y Z y z p sx. cond_expec_2 X Y Z y z p sx =
  ∑_space sx (λx. x × Normal (cond_prob p (PREIMAGE X {x} ∩ p_space p) (PREIMAGE Y {y} ∩
  p_space p ∩ (PREIMAGE Z {z} ∩ p_space p)))).
```

where the HOL function `Normal` is used to convert a real value to its corresponding value in an extended real. Based on the above definition, we can easily verify, in HOL, that $E(X \mid Y = y) = E(X \mid Y = y, \mathbb{1}_{\Omega} = 1)$, where $\mathbb{1}_{\Omega}$ is the indicator function on the probability space Ω.

- The conditional expectation of a function of a random variable is formally verified in HOL as

$$E(g(X) \mid Y = y) = \sum_{x} g(x) \times Pr(X = x \mid Y = y) \tag{12}$$

- The law of total expectation: By analogy to the law of total probability (Eq. 7), we formally verify that

$$E(X) = \sum_{y} E(X \mid Y = y) \times Pr(Y = y) \tag{13}$$

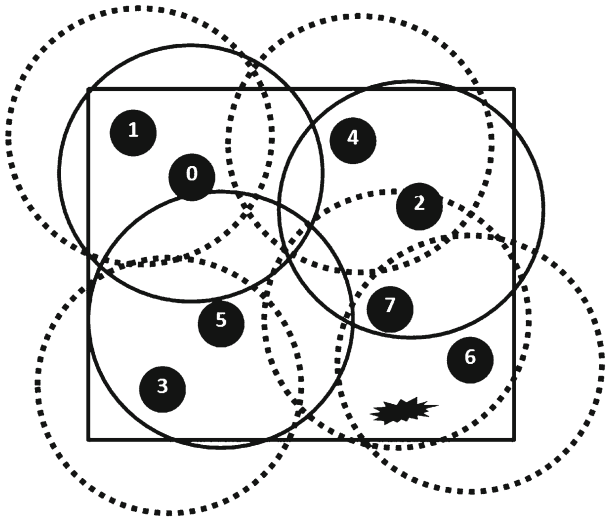


**Fig. 1.** An example of the k-set randomized scheduling for 8 nodes and 2 subsets

### 3.3. The k-set randomized scheduling algorithm

The k-set randomized scheduling has been separately proposed by [AGP04, Liu04]. The main idea of this algorithm can be summarized as follows. Consider a WSN that is formed by randomly deploying $n$ sensor nodes over a two-dimensional field of interest. Every sensor can only sense the environment and detect events within its circular sensing area. During the initialization phase, the k-set randomized scheduling is run on every node as follows. Each node starts by randomly picking a number ranging from 0 to $(k-1)$. We designate the selected number by $i$. Now, the node is assigned to the sub-network $i$, denoted by $S_i$, and will be turned on only within the working time slot $T_i$ of that subset. During the other time slots, it will be in the idle state. Hence, during the time slot $T_i$, only the nodes belonging to the subset $S_i$ will be active and can detect an occurring event. The scheduling algorithm terminates by creating $k$ disjoint sub-networks that work independently and alternatively so that the energy over the whole network can be preserved. Intuitively, when the wireless sensor network is quite dense, each subset alone can cover most of the area. Finally, it is important to note that each node joins a single subset with the same probability $\left(\frac{1}{k}\right)$ since nodes are uniformly and independently deployed over the area of interest.

For illustration purposes, Fig. 1 shows how the k-set randomized scheduling algorithm splits arbitrarily a small WSN of eight sensor nodes to two sub-networks. The eight nodes, randomly deployed in the monitored region, are identified by IDs ranging from 0 to 7. The two sub-networks are denoted $S_0$ and $S_1$. Each node randomly chooses a number 0 or 1 in order to be assigned to one of these two sub-networks. Suppose that nodes 0; 2; 5, select the number 0 and join the subset $S_0$ and nodes 1; 3; 4; 6; 7, choose the number 1 and join the subset $S_1$. These two sub-networks will work alternatively, i.e., when the nodes 0; 2; 5, with sensing ranges denoted by the solid circles, are active, the nodes 1; 3; 4; 6; 7, illustrated by the dashed circles, will be idle and vice-versa.

## 4. Formalization of the detection properties

In a wireless sensor network, an occurring event of any length is expected to be detected with a given probability by one or more active nodes within a given delay. The detection behavior of the network is hence a key feature whose performance is measured through two widely used metrics which are the detection probability and the detection delay [YMG08]. In this section, we first formally reason about some properties related to the intrusion period of any occurring event. Next, we exploit this analysis to develop the higher-order-logic formalization of the main detection metrics in WSNs using the k-set randomized scheduling.

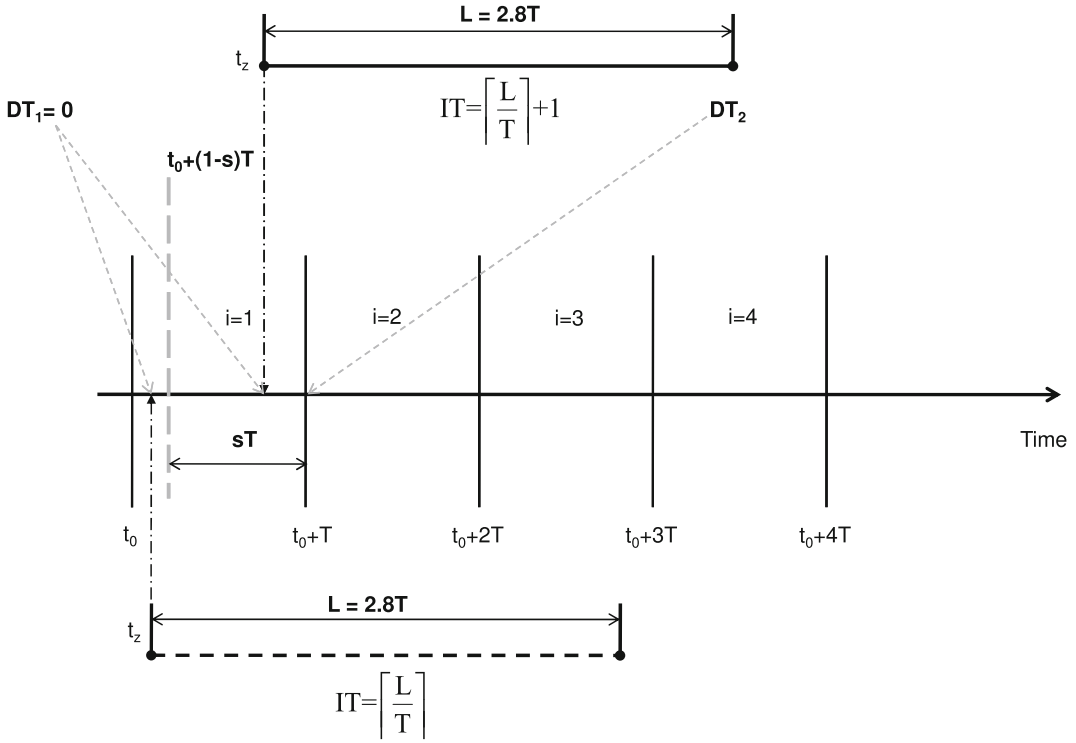


**Fig. 2.** Detection analysis [XCW+10]

### 4.1. Formalization of the intrusion period

According to the description of the k-set randomized algorithm, given in Sect. 3, the $k$ formed subsets of nodes $\{S_i, 1 \leq i \leq k\}$ are disjoint and work alternatively within their scheduling time cycles/slots $\{T_i, 1 \leq i \leq k\}$. In a wireless sensor network, an event, e.g., the outbreak of a fire in a forest, happens randomly at any time. The duration of this event, denoted $L$, will obviously overlap with a number of scheduling cycles $T$ (see Fig. 2). We are interested in formally verifying the average number of overlapping cycles with an intrusion period $L$.

According to the analysis done in [XCW+10], the number of overlapping cycles with an intrusion event depends mainly on $s$, which is the remainder of the intrusion period $L$ in terms of the number of slots $T$. Let $t_0$ be any reference time and $t_z$ the beginning of the intrusion event. Figure 2 shows how the interval $[t_0, t_0 + T]$ is split into two regions according to $s$. Hence, if $t_z$ belongs to the interval

- $[t_0, t_0 + (1 - s) \times T]$, then $L$ overlaps $\lceil \frac{L}{T} \rceil$ with the probability $(1 - s)$.
- $]t_0 + (1 - s) \times T, T[$, then $L$ overlaps $\left(\lceil \frac{L}{T} \rceil + 1\right)$ with the probability $s$.

More specifically, by expressing $L$ in terms of $T$, the variable $s$ has been defined by the following Eq. [XCW+10].

$$s = \frac{L}{T} + 1 - \left\lceil \frac{L}{T} \right\rceil \tag{14}$$

As an example, let us take an intrusion event which lasts for a duration $L = 2.8T$, as illustrated in Fig. 2. Hence, $L$ overlaps either $\lceil \frac{L}{T} \rceil = \lceil \frac{2.8T}{T} \rceil = 3$ cycles with the probability $(1 - s = 0.2)$, or 4 cycles with the probability $(s = 0.8)$.

We can now formalize in higher-order logic the average number of overlapping cycles with an intrusion period $L$. For this purpose, we proceed by first formally specifying the corresponding random variable which describes the number of overlapping cycles within an intrusion period $L$. Based on the above description, we model this behavior by a random variable denoted by $IT$. This random variable can be characterized in higher-order logic by the following predicate `intr_distr_rv` that accepts five parameters: $IT$: a random variable that returns an

extended real number, $p$: the probability space, $s$: the variable specified in Eq. (14), $L$: the length of the intrusion period, and $Ts$: the length of a time slot. Please note that for the sake of simplicity, we take $s$ as a separate variable, although it depends only on $L$ and $Ts$.

**Definition 4.1**

```
⊢ ∀IT p s (L:real) (Ts:real). intr_distr_rv IT p s L Ts =
                              (real_random_variable IT p) ∧
                              (IMAGE IT (p_space p) = {⌈L/Ts⌉; ⌈L/Ts⌉ + 1}) ∧
                              (distribution p IT {⌈L/Ts⌉} = 1 - s).
```

The definition above specifies $IT$ as a real random variable on the probability space $p$ such that the image of $IT$ on `(p_space p)` is in $\{\lceil \frac{L}{Ts} \rceil; \lceil \frac{L}{Ts} \rceil + 1\}$, and its probability distribution over $\{\lceil \frac{L}{Ts} \rceil\}$ is $(1 - s)$.

Next, we formally verify, in Theorem 4.1, the average number of overlapping cycles with an intrusion period $L$, which is the expectation of the random variable $IT$.

**Theorem 4.1**

```
⊢ ∀IT p s L Ts. (0 < Ts) ∧ (0 < L) ∧ (intr_distr_rv IT p s L Ts)
  ⇒ (expectation p IT = Normal(L/Ts + 1))).
```

where the function `expectation`, used in the above theorem, designates the higher-order-logic formalization of the expectation of a random variable that returns an extended real, whereas, the HOL function `Normal` is used to convert a real value to its corresponding value in an extended real. The proof of Theorem 4.1 is based on the verification of the probability distribution on $\{\lceil \frac{L}{Ts} \rceil\}$ and $\{(\lceil \frac{L}{Ts} \rceil + 1)\}$, along with some analysis on extended real.

## 4.2. Formalization of the detection probability

The probability of detecting an intrusion event (D) is usually specified using the probability of the event "being unable to detect an intrusion (UD)" [XZSC07, XCW+10]. Thus, using the probability rule of complement, we have:

$$Pr(D) = 1 - Pr(UD) \tag{15}$$

The detection performances of a wireless sensor network mainly depends on the number of nodes covering the occurring events. In [EHTA11], we demonstrated that the number of nodes covering a point where the intrusion event happens is a Binomial random variable (c) with the following probability.

$$Pr(c = j) = C_n^j \times \left(\frac{r}{a}\right)^j \times \left(1 - \left(\frac{r}{a}\right)\right)^{n-j} \tag{16}$$

where $C_n^j$ is the binomial coefficient indexed by the number $j$ of nodes covering an occurring event and the total number $n$ of deployed nodes. The parameters $r$ and $a$, used in Eq. (16), are the size of the sensing area of each sensor and the size of the monitored area, respectively, and $\left(\frac{r}{a}\right)$ is the probability that each sensor covers a given point. The Binomial random variable with $n$ trials and success probability $q = \left(\frac{r}{a}\right)$ is specified in the following definition [EHTA11].

**Definition 4.2**

```
⊢ ∀X p q n. binomial_distr_rv X p q n = (real_random_variable X p) ∧
            (IMAGE X (p_space p) = IMAGE (λx.&x) (count (SUC n))) ∧
            (∀m. &m IN (IMAGE X (p_space p)) ⇒
            (distribution p X {&m} = &(binomial n k) × q^k × (1 − q)^(n−k))).
```

where $X$ is a real random variable defined on the probability space $p$, and `IMAGE (λx.&x) (count (SUC n))` generates the support of the Binomial, while the operator `&` allows the conversion of the natural number $m$ into its extended number counterpart. The function `binomial`, used in the above definition, is the higher-order-logic formalization of the binomial coefficient for reals, which we specified in HOL in Definition 4.3.

**Definition 4.3**

```
⊢ ∀n k. binomial n k = (binomial n 0 = (1:num)) ∧
                       (binomial 0 (SUC k) = (0:num)) ∧
                       (binomial (SUC n) (SUC k) = binomial n (SUC k) + binomial n k).
```

Given that the events $\{c = j, 0 \leq j \leq n\}$ form a partition of the entire sample space (`Ω = p_space p`), we can establish from Eq. (15), using the law of total probability (Eq. 7), that

$$Pr(D) = 1 - \sum_{j=0}^{n} Pr(UD \mid c = j) \times Pr(c = j) \tag{17}$$

where $Pr(UD \mid c = j)$ is the conditional probability of being unable to detect the intrusion event given that $(c = j)$.

Based on the analysis done in [XCW+10], we discuss the probability $Pr(UD \mid c = j)$ according to the values of $j$, i.e., the number of sensor nodes covering a point when the intrusion event happens, and $L$, i.e., the intrusion period.

- **Case 1.** $(j = 0)$ and for any duration $L$, $Pr(UD \mid c = 0) = 1$.
  Given that there is 0 covering nodes, it is sure that an intrusion event can never be detected.
- **Case 2.** $\{0 < j \leq n\} \cap \{L \geq (k-1) \times T\}$, $Pr(UD \mid c = j) = 0$.
  Since there are $k$ working rounds, each of length $T$, an event lasting more than $(k-1) \times T$, and having at least one covering active node $(0 < j)$ will be always detected.
- **Case 3.** $\{0 < j \leq n\} \cap \{L < (k-1) \times T)\}$, $Pr(UD \mid c = j) \neq 0$. An event lasting less than $(k-1) \times T$ with at least one covering active node $(0 < j)$, will be usually detected with a given probability which is not null.

By extracting the first term (j = 0) of the summation in Eq. (17), we obtain

$$Pr(D) = 1 - (Pr(UD \mid c = 0) \times Pr(c = 0) + \sum_{j=1}^{n} Pr(UD \mid c = j) \times Pr(c = j)) \tag{18}$$

According to case 1, we have $Pr(UD \mid c = 0) = 1$, and we hence can rewrite Eq. (18), using Eq. (16), as

$$Pr(D) = 1 - ((1-q)^n + \sum_{j=1}^{n} Pr(UD \mid c = j) \times Pr(c = j)) \tag{19}$$

In the following, we are interested in formally verifying the detection probability $Pr(D)$ for occurring events of any length $L$. More particularly, we focus on the formalization of the summation term of Eq. (19). For that purpose, we distinguish two cases, i.e., $\{L < (k-1) \times T)\}$ and $\{L \geq (k-1) \times T)\}$.

### *4.2.1. Detection probability for events such that* $\{L < (k-1) \times T)\}$.

The mathematical model for the performance analysis of the detection probability has directly given the final result of Eq. (19). Only few explanations related to pure mathematical steps can be found in [XZSC07]. However, in order to achieve accurately the higher-order-logic formalizations of Eq. (19), we require to reason about all the implicit steps related to the probabilistic analysis.

According to the intrusion period analysis, done in Sect. 4.1, we know that the intrusion period $L$ may overlap either $\lceil \frac{L}{T} \rceil$ or $(\lceil \frac{L}{T} \rceil + 1)$ scheduling cycles $T$. Thus, an intrusion event which lasts $L$, cannot be detected either when $L$ overlaps $\lceil \frac{L}{T} \rceil$ cycles, or when $L$ overlaps $(\lceil \frac{L}{T} \rceil + 1)$ cycles. Using the following events

- $A_{12}$ = The intrusion period $L$ overlaps $\lceil \frac{L}{T} \rceil$ cycles.
- $A_{22}$ = The intrusion period $L$ overlaps $(\lceil \frac{L}{T} \rceil + 1)$ cycles.

It is possible to express the whole event of non-detection, denoted by $UD$, as follows

$$UD = UD \cap (A_{12} \cup A_{22}) \tag{20}$$

Now, applying Eqs. (4) and (6) to $Pr(UD \mid c = j)$ in Eq. (19), along with the fact that the events $A_{12}$ and $A_{22}$ are disjoint, we get the following result.

$$\begin{aligned} Pr(UD \mid c = j) = {} & Pr(UD \mid A_{12} \cap (c = j)) \times Pr(A_{12} \mid c = j) \\ & + Pr(UD \mid A_{22} \cap (c = j)) \times Pr(A_{22} \mid c = j) \end{aligned} \tag{21}$$

Intuitively, for a given intrusion event of length $L$, the occurrence of the event ($A_{12}$ = L overlaps $\lceil \frac{L}{T} \rceil$ cycles), and the event $(c = j)$ describing that there are $j$ covering nodes, are governed by distinct and noninteracting physical processes [Fel68]. Hence, the two events turn out to be independent. According to Eq. (3), we get hence $Pr(A_{12} \mid c = j) = Pr(A_{12}) = Pr(IT = \lceil \frac{L}{Ts} \rceil)$, where $IT$ is the intrusion random variable as specified in Definition 4.1. Similarly, we obtain $Pr(A_{22} \mid c = j) = Pr(IT = \lceil \frac{L}{Ts} \rceil + 1)$. This allows us to rewrite the RHS of Eq. (21) as

$$Pr(UD \mid A_{12} \cap (c = j)) \times Pr(A_{12}) + Pr(UD \mid A_{22} \cap (c = j)) \times Pr(A_{22}) \tag{22}$$

On the other hand, the event "$UD \mid A_{12} \cap (c = j)$" indicates the event of "being unable to detect an intrusion event" given that "the intrusion period $L$ overlaps $\lceil \frac{L}{T} \rceil$ cycles" and "there are $j$ covering nodes". Indeed, if an event, covered with $j$ nodes and overlapping $\left(h = \lceil \frac{L}{T} \rceil\right)$ rounds, is not detected, then it means that all the $j$ covering nodes miss the $h$ consecutive subsets. In other words, the sequence of $h$ subsets do not contain covering nodes. Such event is expressed by the following equation.

$$B_{h,c} = H_{1,c} \cap H_{2,c} \cap .. \cap H_{i,c} \cap .. \cap H_{h,c} = \left( \bigcap_{i=1}^{h} H_{i,c} \right) \tag{23}$$

where $H_{i,c}$ is the event that none of the $c$ covering sensor nodes belongs to the working subset $i$, i.e., $H_{i,c}$ is empty, and the set of events $\{H_{1,c}, H_{2,c}, ..., H_{h,c}\}$ is mutually independent. We say that a finite set of events is mutually independent if and only if every event is independent of any intersection of the other events [Fel68]. The probability of the above event (Eq. 23) has been proved in [EHTA11], to be equal to $\left(\frac{k-h}{k}\right)^c$, where $k$ is the number of disjoint subsets.

Accordingly, Eq. (19) becomes

$$Pr(D) = 1 - ((1-q)^n + \sum_{j=1}^{n} [Pr(A_{12}) \times Pr(B_{\lceil \frac{L}{T} \rceil, j}) + Pr(A_{22}) \times Pr(B_{\lceil \frac{L}{T} \rceil + 1, j})]) \tag{24}$$

Based on the above reasoning, we successfully verify, in Theorem 4.2, the final expression of the detection probability $Pr(D)$ for events lasting $\{L < (k-1) \times T\}$.

**Theorem 4.2** ⊢ ∀p X IT UD_rv k q n s L Ts. (prob_space p) ∧
(1 < k) ∧ (1 ≤ n) ∧ (0 < q < 1) ∧ (sn_covers_p X p q n) ∧ (0 < Ts) ∧
(0 < L) ∧ (L < &(k-1)×Ts) ∧ (0 < s < 1) ∧ ((udset n k s L Ts q) ∈ events p) ∧
(intr_distr_rv IT p s L Ts) ∧ (sbst_empty_sch_rv (UD_rv (SUC i)) p k c (SUC i)) ∧
(indep_rv p IT X Borel Borel) ∧
(cond_prob p (udset n k s L Ts q) (PREIMAGE X {0} ∩ p_space p) = 1) ∧
(A12 = PREIMAGE IT {⌈L/Ts⌉} ∩ p_space p) ∧ (A22 = PREIMAGE IT {⌈L/Ts⌉ + 1} ∩ p_space p) ∧

```
(Hic = IMAGE (λi. PREIMAGE (UD_rv (SUC i)) {1} ∩ p_space p)) ∧
(∀x. x ∈ count (SUC n)) ⇒
(cond_prob p (udset n k s L Ts q) (A12 ∩ (PREIMAGE X {&x} ∩ p_space p)) =
prob p (⋂_(i<⌈L/Ts⌉) Hic) ∧
(cond_prob p (udset n k s L Ts q) (A22 ∩ (PREIMAGE X {&x} ∩ p_space p)) =
prob p (⋂_(i<⌈L/Ts⌉+1) Hic)))
⇒ (prob p (p_space p DIFF (udset n k s L Ts q)) =
1 − (1 − s) × (1 − (⌈L/Ts⌉)/k × q)^n − s × (1 − (⌈L/Ts⌉+1)/k × q)^n).
```

where

- `sn_covers_p` is the Binomial random variable (Definition 4.2).
- `intr_distr_rv` is the intrusion random variable (Definition 4.1).
- `sbst_empty_sch_rv` is the higher-order-logic formalization of an empty sub-network in HOL. We modelled such behavior by a Bernoulli random variable with success probability $\left(1 - \frac{1}{k}\right)^c$, and the corresponding HOL function is as follows [EHTA11]

```
⊢ ∀X p pr. bernoulli_distr_rv X p pr = (real_random_variable X p) ∧
                                       (IMAGE X (p_space p) = {0;1} ∧
                                       (distribution p X {1} = pr).
```

- The assumption `(indep_rv p IT X Borel Borel)` ensures the independence between the two random variables $X$ and $IT$.
- The HOL function `(udset n k s L Ts q)` models the main event of non-detection $UD$, as specified in Eq. (15). This function depends on various design parameters, i.e., $n$: the number of sensor nodes, $k$: the number of sub-networks, $L$: the intrusion period, $Ts$: the scheduling time slot, and $s$: the remainder of $L$ in terms of $Ts$.
- The assumption `(cond_prob p (udset n k s L Ts q) (PREIMAGE X {0} ∩ p_space p) = 1)` reflects the first case, discussed at the beginning of this subsection.
- The events `A12`, `A22`, and `Hic` are the HOL formalizations of the same events used throughout our mathematical reasoning.
- The last assumption is the probability equality discussed just after Eq. (22).
- The event `((p_space p) DIFF (udset n k s L Ts q))` formalizes the complement event of $UD$.

The proof of the above theorem is primarily based on the application of the total probability law (Eq. 7) which further requires the verification of the corresponding assumptions regarding the partition of the events (Theorem 3.1). Moreover, various conditional probability rules (Eqs. 3, 4, 5, 6 and 7), have been used as well. For that purpose, the proof utilizes the measurability of the different events and the verification of the probability distributions of the events $A_{21}$ and $A_{22}$, and a lot of real analysis. In particular, a considerable amount of real analysis related to Theorem 4.3 formalizing the Binomial theorem for reals, and to the summation function has been necessary to achieve this proof.

**Theorem 4.3**

```
⊢ ∀(a:real) (b:real) n. (a + b)^n = ∑_0^n (λi. &(binomial n i) × a^(n−i) × b^i).
```

### *4.2.2. Detection probability for events such that* $\{L \geq (k-1) \times T\}$.

According to the second case, discussed at the beginning of this subsection, we simply verify that the detection probability $Pr(D)$ is equal to

$$Pr(D) = 1 - (1 - q)^n \tag{25}$$

using Theorem 4.2. Such result is very significant since it illustrates the linking between our coverage formalizations, done in [EHTA11], and the new results on the detection probability $Pr(D)$. In general, a point in the area

is covered if any occurring event at this point can be detected. Such feature is measured through the network coverage intensity $C_n$, which determines how well the monitored area is covered [LWXS06]. When an event lasts for a duration ($L \geq (k-1) \times T$), it means that a full working cycle, lasting $k \times T$, is spent at least one time, and all the sub-networks $\{S_i, 0 \leq i \leq n\}$ have been hence working at least once. The intuition is that such event is surely detected within one of the working subsets, and its detection probability is equal to the coverage measurement of the network, when the whole network is assimilated to one sub-network, i.e, $C_n$ for $(k = 1)$. The above equation formally confirms this intuition, and shows how the behavior of the detection probability $Pr(D)$ for events lasting $(L \geq (k-1) \times T)$ matches the one for network coverage intensity $C_n$ for $(k = 1)$.

### 4.3. Formalization of the average detection delay

Within a wireless sensor network, the average detection delay is generally defined as the expectation of the time elapsed from the occurrence of an intrusion event to the time when this event is detected by some sensor nodes [XCW+10, LWXS06]. In this part, we target the formal verification of this average detection delay, denoted by $E(D)$. Mathematically, $E(D)$ is specified as the expectation of the random variable $D$ describing the detection delay. We suppose that $E(D)$ is finite.

Let $DT_i$ the average time that the intrusion is detected in the $i^{th}$ round. For the first round $(i = 1)$, the delay is obviously zero $(DT_1 = 0)$. Since the subsets of nodes are working by rounds (cf. Fig. 2), it is thus intuitive that the delay for detecting an intrusion depends on the detection round $i$. In addition, the $DT_i$ values depend also on the starting time, $t_z$, of the intrusion, i.e., $A_{12}$ and $A_{22}$. Hence, for the second round $(i = 2)$, based on Fig. 2, we can find that

- If $t_z \in [t_0, t_0 + (1-s) \times T]$, then $(DT_2 = T - \frac{(1-s) \times T}{2})$.
- If $t_z \in ]t_0 + (1-s) \times T, T[$, then $(DT_2 = \frac{s \times T}{2})$.

More generally, according to the original specification [XCW+10, LWXS06], if $t_z \in [t_0, t_0 + (1-s) \times T]$, i.e., given $A_{12}$, then:

$$DT_i \mid A_{12} = \begin{cases} 0 & \text{if } i = 1 \\ \left((i-1) - \frac{(1-s)}{2}\right) \times T & \text{if } 1 < i \leq \lceil \frac{L}{T} \rceil \end{cases} \tag{26}$$

However, when $t_z \in ]t_0 + (1-s) \times T, T[$, we have

$$DT_i \mid A_{22} = \begin{cases} 0 & \text{if } i = 1 \\ \left((i-2) + \frac{s}{2}\right) \times T & \text{if } 1 < i \leq \lceil \frac{L}{T} \rceil + 1 \end{cases} \tag{27}$$

Note that the notations $(DT_i \mid A_{12})$ and $(DT_i \mid A_{22})$ refer to the values taken by the random variable $D$ given $A_{12}$ and $A_{22}$, respectively.

Based on Eqs. (26) and (27), we notice how the detection delay values depend on the detection round $i$. Consider the random variable $DR_i$ that describes the detection round. Conditioning on the events $A_{12}$ and $A_{22}$, the values of $DR_i$ are

$$DR_i \mid A_{12} = \{i + 1 \mid 0 \leq i \leq ph1 - 1\} \text{ where } ph1 = min\left(k, \left\lceil \frac{L}{T} \right\rceil\right) \tag{28}$$

$$DR_i \mid A_{22} = \{i + 1, 0 \leq i \leq ph2 - 1\} \text{ where } ph2 = min\left(k, \left\lceil \frac{L}{T} \right\rceil + 1\right) \tag{29}$$

The minimum values for the variables $ph1$ and $ph2$ are considered since we have at most $k$ detection rounds (cf. Fig. 2). As an example, consider a WSN which is randomly scheduled into $(k = 3)$ sub-networks, and two intrusion events $E1$ and $E2$ whose starting time $t_z$ is in $[t_0, t_0 + (1-s) \times T]$, and lasting $(L1 = 1.8 \times T)$ and $(L2 = 3.2 \times T)$, respectively. In the case of event $E1$, $\lceil \frac{L1}{T} \rceil = 2$, and the possible rounds of detection would be

$i = \{1, 2\}$. For event $E2$, $\lceil \frac{L2}{T} \rceil = 4$, but the potential detection rounds are $i = \{1, 2, 3\}$, i.e., at most 3 which is equal to $k$.

According to the two above equations, we formally define a general HOL function that describes the detection round random variable in Definition 4.4.

**Definition 4.4**

```
⊢ ∀DR p ph. delay_rnd_rv DR p ph =
                        (real_random_variable DR p) ∧
                        (IMAGE DR (p_space p) = IMAGE (λj. &SUC j) (count ph)).
```

The main expected detection delay $E(D)$ has been formalized in HOL using the function `delay_wsn`, which is specified as follows

**Definition 4.5** `⊢ ∀p D n k q. delay_wsn p D n k q = expectation p (D n k q).`

where $p$ is the probability space, $D$ is a random variable, $n$ is the number of deployed nodes, $k$ is the number of disjoint subsets, and $q$ is the probability that each sensor covers a given point. The expected detection delay $E(D)$ can be mathematically written, using the total expectation law (Eq. 13) and Eq. (16), as

$$\begin{aligned} E(D) &= \sum_{j=1}^{n} E(D \mid c = j) \times Pr(c = j) \\ &= \sum_{j=1}^{n} E(D \mid c = j) \times C_n^j \times \left(\frac{r}{a}\right)^j \times \left(1 - \left(\frac{r}{a}\right)\right)^{n-j} \end{aligned} \tag{30}$$

where $E(D \mid c = j)$ is the conditional expectation of the real random variable $D$ with respect to the event $(c = j)$. Notice that the case $(c = 0)$ is not considered in Eq. (30). Indeed, if there is no covering node, then an intrusion can never be detected, and the delay $E(D)$ will be infinite which is not desirable.

In higher-order logic, we model the detection delay behavior, in Definition 4.6, as a real random variable with a finite image on the space $\Omega$.

**Definition 4.6**

```
⊢ ∀D p. delay_rv D p = (real_random_variable D p) ∧
                       FINITE (IMAGE D (p_space p)).
```

In the following, we focus on the formal verification of the term $E(D \mid c = j)$ in Eq. (30) for occurring events of any length $L$. Based on the definition of conditional expectation (Eq. 10), $E(D \mid c = j)$ can be mathematically expressed as

$$E(D \mid c = j) = \sum_{d} (D = d) \times Pr(D = d \mid c = j) \tag{31}$$

Applying the total probability law (Eq. 7) on the partition $\{A_{12}, A_{22}\}$, and given the independence of the random variable $IT$ and $c$ (Eq. 3), we can establish, using Eq. (6), that

$$\begin{aligned} E(D \mid c = j) = (1 - s) \times \sum_{d} (D = d) \times Pr(D = d \mid A_{12} \cap (c = j)) + \\ s \times \sum_{d} (D = d) \times Pr(D = d \mid A_{22} \cap (c = j)) \end{aligned} \tag{32}$$

The RHS of Eq. (32) can be now rewritten, using the reverse definition of conditional expectation for two events (Eq. 11), as

$$(1-s) \times E(D \mid A_{12}, (c=j)) + s \times E(D \mid A_{22}, (c=j)) \tag{33}$$

Based on the above equation, we can clearly distinguish two distinct conditional expectations given the events $A_{12}$ and $A_{22}$. According to the analysis done at the beginning of this subsection, these conditional expectations can be established as

$$E(D \mid A_{12}, (c=j)) = E(DC1 \mid c=j) \tag{34}$$

$$E(D \mid A_{22}, (c=j)) = E(DC2 \mid c=j) \tag{35}$$

where $DC1$ and $DC2$ are the random variables describing the detection delay when ($A_{12} = L$ overlaps $\lceil \frac{L}{T} \rceil$ cycles) and ($A_{22} = L$ overlaps ($\lceil \frac{L}{T} \rceil + 1$) cycles), respectively. More specifically, based on Eqs. (26) and (27), $DC1$ and $DC2$ can be written as

$$DC1 = \left(\lambda\, x. \left(x - \frac{3}{2} + \frac{s}{2}\right) \times T\right) \circ DR1 \tag{36}$$

$$DC2 = \left(\lambda\, x. \left(x - 2 + \frac{s}{2}\right) \times T\right) \circ DR2 \tag{37}$$

where the $\circ$ operator denotes the function composition, and $DR1$ and $DR2$ are the delay round random variables given $A_{12}$ and $A_{22}$, respectively, as described in Eqs. (28) and (29).

Plugging the above two equations, into Eqs. (34) and (35), and applying the conditional expectation of a function of a random variable (Eq. 12), we derive, from Eq. (33), that the conditional expectation of $D$ given $(c=j)$, $E(D \mid c=j)$, equals

$$\begin{aligned}(1-s) \times \sum_{i=2}^{ph1} \left(i - \frac{3}{2} + \frac{s}{2}\right) \times T \times Pr(DR1 = i \mid A_{12} \cap (c=j)) \\ + s \times \sum_{i=2}^{ph2} \left(i - 2 + \frac{s}{2}\right) \times T \times Pr(DR2 = i \mid A_{22} \cap (c=j))\end{aligned} \tag{38}$$

Now, analyzing the relationship between the random variables, we can establish that $DR1$ and $IT$ are conditionally independent given the random variable $c$. Indeed, in terms of events, the information $A_{12}$ does not add anything about $(DR1 = i)$ if we already know that $(c=j)$. Similarly for $(DR2 = i)$ and $A_{22}$ given $(c=j)$. Using Eq. (8), we can simplify Eq. (38) into

$$\begin{aligned}E(D \mid c=j) = (1-s) \times \sum_{i=2}^{ph1} \left(i - \frac{3}{2} + \frac{s}{2}\right) \times T \times Pr(DR1 = i \mid c=j) \\ + s \times \sum_{i=2}^{ph2} \left(i - 2 + \frac{s}{2}\right) \times T \times Pr(DR2 = i \mid c=j)\end{aligned} \tag{39}$$

Developing the terms $Pr(DR1 = i \mid c = j)$ and $Pr(DR2 = i \mid c = j)$, in the above equation, according to the definition of conditional probability (Eq. 2) along with Eq. (7), we get the following result.

$$E(D \mid c = j) = (1 - s) \times \sum_{i=2}^{ph1} \frac{(i - \frac{3}{2} + \frac{s}{2}) \times T \times Pr((DR1 = i) \cap (c = j))}{\sum_{i=1}^{ph1} Pr((DR1 = i) \cap (c = j))} + s \times \sum_{i=2}^{ph2} \frac{(i - \frac{3}{2} + \frac{s}{2}) \times T \times Pr((DR2 = i) \cap (c = j))}{\sum_{i=1}^{ph2} Pr((DR2 = i) \cap (c = j))} \tag{40}$$

We formally verify, in Theorem 4.4, the HOL theorem formalizing Eq. (40).

**Theorem 4.4**

```
⊢ ∀p X D n q IT s L Ts DC1 DC2 DR1 DR2 ph1 ph2.
  (prob_space p) ∧ (events p = POW (p_space p)) ∧ (delay_rv D p) ∧ (1 < k) ∧
  (0 < q < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (0 < s < 1) ∧ (intr_distr_rv IT p s L Ts) ∧
  (sn_covers_p X p q n) ∧ (indep_rv p IT X Borel Borel) ∧ (1 < ph1) ∧ (1 < ph2) ∧
  (delay_rnd_rv DR1 p ph1) ∧ (delay_DC_rv DC1 DR1 p 3/2 s Ts) ∧
  (delay_rnd_rv DR2 p ph2) ∧ (delay_DC_rv DC2 DR2 p 2 s Ts) ∧
  (cond_indep_rv p DR1 IT X Borel Borel Borel) ∧
  (cond_indep_rv p DR2 IT X Borel Borel Borel) ∧
  (∀i. (1 ≤ i) ∧ (i < SUC n) ⇒
  ((cond_expec_2 D IT X ⌈L/T⌉ (&i) p Dsx = cond_expec_2 DC1 IT X ⌈L/T⌉ (&i) p DC1sx) ∧
  (cond_expec_2 D IT X (⌈L/T⌉ + 1) (&i) p Dsx =
  cond_expec_2 DC2 IT X (⌈L/T⌉ + 1) (&i) p DC2sx)))
  ⇒ (∀i. (1 ≤ j) ∧ (j < SUC n) ⇒
  E(D | c = j) = (1 − s) × ∑ph1 i=2 ((i−3/2+s/2)×Ts×Pr((DR1=i)∩(c=j))) / (∑ph1 i=1 Pr((DR1=i)∩(c=j)))
  +s × ∑ph2 i=2 ((i−2+s/2)×Ts×Pr((DR2=i)∩(c=j))) / (∑ph2 i=1 Pr((DR2=i)∩(c=j))).)
```

where

- The assumptions `(cond_indep_rv p DR1 IT X Borel Borel Borel)` and `(cond_indep_rv p DR2 IT X Borel Borel Borel)` ensure the conditional independence between the different random variables.
- The variables `DC1` and `DC2`, as described in Eqs. (36) and (37), are characterized through the HOL function `(delay_DC_rv DC DR p a s Ts)` which is defined as follows

```
⊢ ∀DC DR p a s Ts. delay_DC_rv DC DR p a s Ts =
                   (∀x. x ∈ (p_space p) ⇒ (0 ≤ DC x)) ∧
                   (DC = ((λx. (x - a + (Normal s)/2) × (Normal Ts))) ∘ DR).
```

- The variable `Dsx = (IMAGE D (p_space p), POW (IMAGE D (p_space p)))`, and the same equality applies to `DC1sx` and `DC2sx` for the corresponding variables `DC1` and `DC2`, respectively.

The proof of Theorem 4.4 is quite similar to the proof of Eq. (40) from Eq. (31). In particular, the reasoning was primarily based on the specification of the above function `(delay_DC_rv DC DR p a s Ts)` by considering only positive values, given that it describes the detection delay behavior which can never be negative. In this case, the terms $(i - \frac{3}{2} + \frac{s}{2})$ and $(i - 2 + \frac{s}{2})$ can be shown to be equal 0 for $(i = 1)$, and the correct summation index of the numerator can be hence proved. Moreover, a lot of reasoning associated with the use of summation including the proof of injectivity for some functions, and real analysis, was also required.

In Eq. (40), the event "$(DR1 = i) \cap (c = j)$" indicates that "the intrusion event is detected in the $i^{th}$ round" and "there are $j$ covering nodes". Indeed, if an event, covered with $j$ nodes, is detected in the $i^{th}$ round, then it means that all the $j$ covering nodes miss the $(i - 1)$ consecutive subsets, and the first covering nodes belong to the subset $i$. Such event is exactly the same as the following event.

$$A_{i,j} = \left(\bigcap_{m=1}^{(i-1)} H_{m,j} \cap \overline{H}_{i,j}\right) = \left(B_{i-1,j} \cap \overline{H}_{i,j}\right) \tag{41}$$

where

- $H_{m,j}$ and $B_{i-1,j}$ are the same events used in Eq. (23).
- the set of events $\{B_{i-1,j}, \overline{H}_{i,j}\}$ is mutually independent.

The probability of the above event (Eq. 41) has been already formally verified in [EHTA11], and is equal to $\left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]$.

At the end, we establish that the final average detection delay $E(D)$ (Eq. 30) is

$$E(D) = \sum_{j=1}^{n} E(D \mid c = j) \times C_n^j \times \left(\frac{r}{a}\right)^j \times \left(1 - \left(\frac{r}{a}\right)\right)^{n-j} \tag{42}$$

where

$$\begin{aligned} E(D \mid c = j) = (1-s) \times \sum_{i=2}^{ph1} \frac{(i - \frac{3}{2} + \frac{s}{2}) \times T \times \left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]}{\sum_{i=1}^{ph1}\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j} \\ + s \times \sum_{i=2}^{ph2} \frac{(i - 2 + \frac{s}{2}) \times T \times \left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]}{\sum_{i=1}^{ph2}\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j)} \end{aligned} \tag{43}$$

It is important to note that, for space constraints, the final HOL theorem for the verification of the main function of the average detection delay `delay_wsn` (Definition 4.5) has been omitted from this paper but an interested reader can access it from [Ell13].

In this section, we detailed the higher-order-logic formalizations of the detection performances of wireless sensor networks using the k-set randomized scheduling. The corresponding HOL code is available at [Ell13]. In the next section, we will demonstrate how the resulting universally quantified theorems greatly facilitate the formal analysis of real-world WSN applications.

## 5. Formal analysis of WSN for border surveillance

Wireless sensor networks have been widely explored for border monitoring applications [ADB+04]. The main goal of a WSN deployed for border monitoring is to continuously detect intruding elements with a high probability and a small delay. These systems are useful for the detection of forces or vehicles in a military context [Hew01], or the prevention of illegal intrusions of migrants or terrorists along a country border. In this context, the potential harsh nature of the field of interest makes a random deployment by air-dropping sensors much more practical. In this section, we are interested in formally analyzing the detection performances of a wireless sensor network deployed for a border monitoring application [XZP+09, SWV+11].

Due to the safety-critical feature of the target application, the deployed WSN has to remain alive as long as possible while ensuring an efficient detection. Nevertheless, as stated in [ADB+04], most of the existing WSNs for border monitoring suffer from lifetime limitations, e.g., a REMBASS sensor node, once deployed, can be functional for 30 days only [Hew01]. In case of using the WSN to monitor terrorist intrusions along a mountainous border, it is obviously not required to monitor the whole area at all times. Thus, we can use the k-set randomized scheduling algorithm to preserve energy in a given border monitoring application [XZP+09]. In the specified application, the nodes have a sensing area $r = 30$, and are deployed into an area of size $a = 10000\ \text{m}^2$, whereas, the success probability $q$ of a sensor covering a point, is $q = \frac{r}{a} = 0.28$.

In the previous section, we analyzed the detection probability $Pr(D)$ according to the intrusion length $L$ by distinguishing 2 cases: $\{L < (k-1) \times Ts)\}$ and $\{L \geq (k-1) \times Ts)\}$. It is important to note that, in the current application analysis, we focus on the first case; $\{L < (k-1) \times Ts)\}$, which reflects transient events, that may not be detected, and is thus the most pertinent part of this analysis. For the other case, i.e., $\{L \geq (k-1) \times Ts)\}$, we have already discussed that the detection probability $Pr(D)$ equals the network coverage, and its asymptotic behavior has been investigated in [EHTA13].

Based on our theoretical development done in the previous section, we now conduct a formal asymptotic analysis of the probabilistic detection and delay based on the parameters $n$ and $k$. For that, we are going to tackle the generic case and then instantiate it for the given border monitoring application. Hence, we simply denote `(prob p (p_space p DIFF (udset n k s L Ts q)))` by `(Pd_wsn p n k s L Ts q)` and `(delay_wsn p D n k q)` as `(D_wsn p D n k q)`. In the context of our application, we basically verify two main properties of interest related to the detection probability of the events of interest and the detection delay. Thus, we easily check in HOL that `(prob p (p_space p DIFF (udset n k s L Ts (0.28))))` equals

$$1-(1-s)\times\left(1-\frac{\left(\lceil\frac{L}{Ts}\rceil\right)}{k}\times(0.28)\right)^{n}-s\times\left(1-\frac{\left(\lceil\frac{L}{Ts}\rceil+1\right)}{k}\times(0.28)\right)^{n} \tag{44}$$

and, the expected detection delay, `(delay_wsn p D n k (0.28))`, is

$$\sum_{j=1}^{n} E(D \mid c=j)\times C_n^j\times(0.28)^j\times(1-(0.28))^{n-j} \tag{45}$$

where $E(D \mid c=j)$ represents the expression specified in Eq. (43). Next, we simply denote Eqs. (44) and (45), by `(Pd_surv p n k s L Ts (0.28))` and `(D_surv p D n k (0.28))`, respectively. It is important to note that, for space constraints, and in all the asymptotic analysis below, we only mention the main mathematical assumptions related to the used variables in the detection probability and delay. Whereas, the complete HOL code for these asymptotic analysis can be found in [Ell13].

Hence, we formally verify that the detection probability is an increasing function of $n$, i.e., a larger $n$ value leads to a better detection probability.

**Lemma 5.1**

```
⊢ ∀p k q s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (L < &(k-1)×Ts) ∧
  (0 < q < 1) ⇒ (mono_incr (λn. Pd_wsn p n k s L Ts q))).
```

where `mono_incr` is the HOL definition of an increasing sequence, which we define as follows.

**Definition 5.1**

```
⊢ ∀f. mono_incr f ⇔ ∀n. f n ≤ f (SUC n).
```

Besides, we formally verify, in Lemma 5.2, that the probability of detecting an intrusion event approaches 1 as the number of deployed nodes becomes very very large.

**Lemma 5.2**

```
⊢ ∀p k q s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (L < &(k-1)×Ts) ∧
  (0 < q < 1) ⇒ lim (λn. Pd_wsn p n k s L Ts q) = 1.
                n→+∞
```

where `lim` is the HOL formalization of limit for real sequences.

Similarly, it is also very useful to investigate the delay behavior of the randomized scheduling. Thus, we formally verify, in Lemma 5.3, that the detection delay $D\_wsn$ starts to be decreasing versus the number of nodes $n$ from a given range, denoted $n_0$. Consequently, $D\_wsn$ becomes smaller when a large number of nodes is deployed. In this case, an intrusion is expected to be detected more quickly, since it is likely that many more covering nodes are deployed in the surrounding area.

**Lemma 5.3**

```
⊢ ∀p k q s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (0 < q < 1)
  ⇒ (mono_decr_range (λn. (real (D_wsn p D n k q)))).
```

where the function `real` is used to convert the detection delay of type extended real to its corresponding real value, and the HOL function `mono_decr_range` is specified in Definition 5.2.

**Definition 5.2**

```
⊢ ∀ f. mono_decr_range f ⇔ (∃n0. ∀n. n ≥ n0 ⇒ f (SUC n) ≤ f n).
```

Based on Lemmas 5.1 and 5.2, we establish that any target detection probability $Pd\_wsn$ can be achieved by increasing the number of deployed nodes $n$, for any values of the input variables $k$, $q$, $s$, $L$, and $Ts$. More specifically, these results can be easily verified for the detection probability, $Pd\_surv$, in the context of the given border monitoring application (Lemmas 5.4 and 5.5).

**Lemma 5.4**

```
⊢ ∀p k s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (L < &(k-1)×Ts)
  ⇒ (mono_incr (λn. Pd_surv p n k s L Ts (0.28)))).
```

**Lemma 5.5**

```
⊢ ∀p k s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (L < &(k-1)×Ts)
  ⇒ lim (λn. Pd_surv p n k s L Ts (0.28)) = 1.
    n→+∞
```

In addition, we reconfirm the result of Lemma 5.3 using Lemma 5.6, i.e., increasing the number of deployed nodes $n$ gives smaller detection delays and thus a better performance of the deployed application.

**Lemma 5.6**

```
⊢ ∀p k s L Ts. (1 < k) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts)
  ⇒ (mono_decr_range (λn. (real (D_surv p D n k (0.28))))).
```

According to Lemmas 5.1 and 5.3, enhancing the detection capacities of the deployed WSN, is possible through the deployment of more nodes. However, random deployment is known to be very costly for most WSN applications. In the context of a WSN using the k-set randomized scheduling, it is usually possible to improve the whole detection capacity of the network by simply updating the number of disjoint subsets $k$ by a suitable value.

Based on the parameter $k$, we perform now an interesting study of the limiting behavior of the detection performances. First, we formally verify, in Lemma 5.7, that a smaller $k$ value induces a larger detection probability $Pd\_wsn$, i.e., $Pd\_wsn$ decreases while increasing the value of $k$.

**Lemma 5.7**

```
⊢ ∀p k q n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (0 < q < 1) ∧
  (∀k. L < &(SUC k)×Ts) ⇒ (mono_decr (λk. Pd_wsn p n k s L Ts q)).
```

where the HOL function `mono_decr` defines a decreasing sequence as follows.

**Definition 5.3**

```
⊢ ∀f. mono_decr f ⇔ ∀n. f (SUC n) ≤ f n.
```

We formally confirm, in Lemma 5.8, that given a number of nodes $n$, the detection probability $Pd\_wsn$ goes to 0 when $k$ becomes very large.

**Lemma 5.8**

```
⊢ ∀p k q n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (0 < q < 1) ∧
  (∀k. L < &(SUC k)×Ts) ⇒ lim (λk. Pd_wsn p n k s L Ts q) = 0.
                          k→+∞
```

Furthermore, we show, in Lemma 5.9, that the detection delay of the randomized scheduling, $D\_wsn$, increases as the value of $k$ increases. In other words, the detection delay $D\_wsn$ increases when the WSN is divided into a quite large number of sub-networks $k$. Indeed, the allocated time slot for each subset would be small, so that the active nodes do not have enough time to detect the occurring intrusion.

**Lemma 5.9**

```
⊢ ∀p q n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (0 < q < 1)
  ⇒ (mono_incr (λk. real (D_wsn p D n k q))).
```

It is important to note that the original proof of the above lemma in [XZSC07, XCW+10] was missing a whole fraction term, which is fortunately positive and thus does not finally affect the validity of the function monotonicity.

Now, it is possible to confirm, in the following 2 lemmas, the validity of the generic results given in Lemmas 5.7 and 5.8 for our WSN application.

**Lemma 5.10**

```
⊢ ∀p k n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (∀k. L < &(SUC k)×Ts)
  ⇒ (mono_decr (λk. Pd_surv p n k s L Ts (0.28))).
```

Consequently, for the border monitoring application, increasing $k$ surely saves more energy, but a significant increase in $k$ may induce several sub-networks, which in turns translates to a poor detection probability (Lemma 5.11).

**Lemma 5.11**

```
⊢ ∀p k n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (∀k. L < &(SUC k)×Ts)
  ⇒ lim (λk. Pd_surv p n k s L Ts (0.28)) = 0.
    k→+∞
```

Similarly, we check in Lemma 5.12, that a significant increase in $k$ leads to larger detection delays, i.e., a poor performance.

**Lemma 5.12**

```
⊢ ∀p n s L Ts. (1 ≤ n) ∧ (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts)
  ⇒ (mono_incr (λk. real (D_surv p D n k (0.28)))).
```

The randomized scheduling is thus a dynamic approach which provides performance adjustments of the deployed WSN application according to the value of $k$.

The randomness in the nodes scheduling approach leads to sub-networks of different sizes with respect to the number of nodes. Obviously, the ideal case arises when the algorithm makes a fair split of the network so that all the subsets have the same size, i.e., the same number of nodes which we denote by $m$. The number of nodes $n$ can be written hence as $k \times m$. In what follows, we closely investigate the asymptotic performance behavior of the k-set randomized algorithm in the case of a *uniform* split of the nodes.

In particular, we successfully verify, in Lemma 5.13, the upper limit of the detection probability $Pd\_wsn$ when $n = k \times m$ and $k$ goes to infinity.

**Lemma 5.13**

```
⊢ ∀p m q s L Ts. (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (∀k. L < &(SUC k)×Ts) ∧
  (0 < q < 1) ⇒ lim (λk. Pd_wsn p (k×m) k s L Ts q) =
                k→+∞
  1 − (1 − s) × e^(−⌈L/Ts⌉) × q × m − s × e^(−(⌈L/Ts⌉+1)) × q × m.
```

The proof of the above lemma is based on the important mathematical result $\lim_{k\to+\infty}(1+\frac{x}{k})^k = e^x$, which we have proved beforehand.

Based on Lemma 5.13, the analysis of the above limit versus various parameters such as the intrusion period $L$, and the number of nodes per subset $m$, is now feasible. We hence verify that when $m$ is very large, the detection probability will surely approach 1. Such result is considered as a second verification of Lemma 5.2 in the specific case where $n = k \times m$.

**Lemma 5.14**

```
⊢ ∀p q s L Ts. (0 < s < 1) ∧ (0 < L) ∧ (0 < Ts) ∧ (∀k. L < &(SUC k)×Ts) ∧
  (0 < q < 1) ⇒ lim (λm. lim (λk. Pd_wsn p (k×m) k s L Ts q)) = 1.
               m→+∞  k→+∞
```

Finally, we show that the above mentioned two results are also valuable for the given application for border surveillance through a simple instantiation of the input parameter $q$ by its value. The corresponding HOL analysis is given in the following 2 lemmas.

**Lemma 5.15**

$$\vdash \forall p\ m\ s\ L\ Ts.\ (0 < s < 1) \wedge (0 < L) \wedge (0 < Ts) \wedge (\forall k.\ L < \&(SUC\ k) \times Ts) \Rightarrow \lim_{k \to +\infty} (\lambda k.\ Pd\_surv\ p\ (k \times m)\ k\ s\ L\ Ts\ (0.28)) = 1 - (1 - s) \times e^{-\lceil \frac{L}{Ts} \rceil} \times (0.28) \times m - s \times e^{-\left(\lceil \frac{L}{Ts} \rceil + 1\right)} \times (0.28) \times m.$$

**Lemma 5.16**

$$\vdash \forall p\ s\ L\ Ts.\ (0 < s < 1) \wedge (0 < L) \wedge (0 < Ts) \wedge (\forall k.\ L < \&(SUC\ k) \times Ts) \Rightarrow \lim_{m \to +\infty} (\lambda m.\ \lim_{k \to +\infty} (\lambda k.\ Pd\_surv\ p\ (k \times m)\ k\ s\ L\ Ts\ (0.28))) = 1.$$

Unlike traditional analysis techniques for the validation of a WSN for border surveillance, using the k-set randomized scheduling algorithm, our approach is much more efficient. Indeed, while paper-and-pencil based analysis or simulation [XZP+09] cannot guarantee the correctness of the scheduling performance results, the reported theorems in this paper are absolutely accurate. This distinguishing feature is due to the inherent soundness of theorem proving and its generic nature, e.g., the detection probability for any given values of $n$ and $k$ can be computed by instantiating Theorem 4.2 with appropriate values. Contrarily, simulation is usually restricted to specific network configurations, while probabilistic model checking is frequently using parameter abstraction in order to cope with the state-space explosion problem. Moreover, for each of the formally verified theorems, the set of required assumptions is clearly stated so there is no doubt about missing a critical assumption. Such aspect can never be ensured in simulation and model checking where many assumptions can be taken into account without explicitly mentioning them.

## 6. Discussion

In this work, we provided a completely rigourous method for the performance evaluation of the randomized scheduling algorithm for WSNs through theorem proving. Indeed, the probabilistic feature of the randomized nodes scheduling algorithm makes its analysis challenging for all possible cases. Since the assignment of the sensor nodes to the $k$ sub-networks is randomly done, it may happen that some of the sub-networks are empty. Moreover, due to the random deployment of nodes, the random scheduling can lead to a situation where certain parts of the area are not monitored at all or simultaneously monitored by many sensors. Rigorous performance evaluation of such algorithm is a non-trivial task, especially given the non-exhaustive nature of traditional performance analysis techniques.

Throughout this paper, we developed the formalizations of the detection properties of wireless sensor networks using the k-set randomized scheduling within the HOL theorem prover. In Sect. 4, we have been able to achieve accurate formalizations of the intrusion period of any occurring event, upon which we have built our formal developments of the detection probability and delay. The practical effectiveness of these higher-order-logic developments, have been then illustrated, in Sect. 5, through analyzing a WSN for border surveillance, using the k-set randomized scheduling algorithm.

Due to the undecidable nature of higher-order logic, the development of the detection properties consumed approximately about 260 man hours and 2,400 lines of code. On the other hand, the formal analysis of our application took only 400 lines of HOL code for the verification of Lemmas 5.1, 5.2, 5.7 and 5.8. Whereas, the proofs of the monotonicity of the detection delay versus the two parameters $n$ and $k$ in Lemmas 5.3 and 5.9 have been quite tedious and long, and took at their own 1500 lines of HOL code. Indeed, given the complexity of the mathematical expressions of the detection delay, the HOL analysis of these 2 lemmas was requiring a lot of real reasoning on the convergence of series and the properties of infinite sums. More specifically, to prove Lemma 5.3, we have been obliged to consider another mathematical solution since the initial paper-and-pencil proof [XCW+10] includes some mathematical aspects which were not available in the HOL theories. In addition, looking for the range from which the detection delay starts to be decreasing versus $n$, was somewhere tricky. Regarding the proof of Lemma 5.9, it has been based on computing the derivative of the corresponding real functions and applying the mean value theorem. Similarly, the proofs of Lemmas 5.13 and 5.14 have been quite lengthy consuming in total 600 lines of HOL code. Indeed, as we previously mentioned, these proofs required

the mathematical theorem $\lim_{k\to+\infty}(1+\frac{x}{k})^k = e^x$, which was missing in HOL. The latter is based on a lot of real analysis associated to the definition of the exponential function as a power series and many properties related to the sequences convergence.

Thanks to the sound support of the probability theory [MHT11] available within the HOL theorem prover, we have been able to provide an accurate formalization of the detection performance of the k-set randomized scheduling through an appropriate modelling of its inherent randomness. Based on the discussion, given in Sect. 2 of this paper, it is clear that other analysis techniques can never have this efficiency. Indeed, previous simulation works are mainly based on pseudo-random modelling. Similarly, compared to probabilistic model checkers, a major novelty provided in this paper is the ability to perform formal and accurate reasoning about statistical properties of the problem. Hence, it was possible to verify the detection delay as a statistical measure. Moreover, the generic nature of theorem proving and the high expressibility of higher-order logic, allows us to set up theorems for any values for the number of nodes $n$, the number of disjoint subsets $k$, the success probability $q$, the intrusion period $L$, and the scheduling time slot $T$. Obviously, such generality can never be achieved by simulation and model checking. Finally, because missing a critical assumption can lead to verification failure within the theorem prover, the current approach is distinguishable by its completeness regarding the minimum set of assumptions.

On the other hand, the formal performance analysis of the detection behavior of the border surveillance application distinctly show the usefulness of the theoretical higher-order-logic developments. Furthermore, such verification enables reliable asymptotic reasoning of the deployed WSN. For example, the missing term in the proof of Lemma 5.9 clearly highlights the main strength of formal methods guaranteeing accurate and complete results. It is also important to note that the presented application is a simple case study illustrating the feasibility, but these results can be valuable for any other WSN application as well.

The above mentioned additional benefits, associated with the theorem proving approach, are attained at the cost of the time and effort spent, while formalizing the randomized scheduling algorithm and formally reasoning about its detection properties, by the user. We believe that the main challenge incurred in our work was to map a probabilistic model of a real WSN algorithm [XCW+10, LWXS06], which is far from a pure mathematical problem, into higher-order logic. Indeed, many difficulties were faced in this work. The mathematical modelling of real-world systems is commonly very intuitive. The initial theoretical model [XCW+10, LWXS06] hence included many hidden steps with few attached explanations either when considering the random variables or when applying the probability rules. We have thus to reason correctly about all missing steps so that we can first understand the flow of the theoretical analysis, and achieve then the higher-order-logic formalizations of the detection attributes. At this stage, a good background on probability and a solid knowledge of the WSN context is usually required for a deep understanding of the probabilistic reasoning. Additionally, the assumptions of the original model are never presented exhaustively, whereas, a complete set is essential for a successful verification. Nevertheless, the fact that we were building on top of already verified probability theory related results helped significantly to keep the amount of proof efforts reasonable.

## 7. Conclusions

This paper presents an approach for the formal analysis of the detection performances of wireless sensor networks using the k-set randomized scheduling to preserve energy. In particular, we formalized the notions of intrusion period, detection probability and delay using the measure theoretic formalization of probability theory in the HOL theorem prover. This formalization allows to formally verify the detection related characteristics of most WSNs using the k-set randomized scheduling. In order to illustrate the practical effectiveness of our foundational results, we utilize them to perform the formal probabilistic analysis of a WSN application for border surveillance. The obtained results are exhaustive and completely generic, i.e., valid for all parameter values; a result which cannot be attained in simulation or probabilistic model checking based approach. Moreover, unlike most of the existing work that focuses on the validation of the functional aspects of WSN algorithms, our work is distinguishable by addressing the performance aspects. Finally, the proposed approach described in this paper can be generalized to tackle the formal analysis of the same randomized scheduling under other assumptions, or even other probabilistic problems in the WSN context. Indeed, the presented formalizations can be valuable for formally verify the same algorithm with, for example, a modified shape of the intrusion object [XZP+09]. In addition, the higher-order-logic formalizations of some common random variables such as Bernoulli or Binomial can be very useful for the formal analysis of any probabilistic analysis problem.

This work lays also an interesting foundation for our future work on the higher-order-logic formalization of the lifetime properties of WSNs using the k-set randomized scheduling. Similarly, once the formal reasoning support of the lifetime aspect is developed in the HOL theorem prover, the performance of other interesting WSN applications, such as underwater monitoring, can also be formally analyzed.

## References


[Abr09] Abrial J (2009) Faultless systems: yes we can! Computer 42(9):30–36

[ADB+04] Arora A, Dutta P, Bapat S, Kulathumani V, Zhang H, Naik V, Mittal V, Cao H, Demirbas M, Gouda M, Choi Y, Herman T, Kulkarni S, Arumugam U, Nesterenko M, Vora A, Miyashita M (2004) A line in the sand: a wireless sensor network for target detection, classification, and tracking. Comput Netw 46(5):605–634

[AGP04] Abrams Z, Goel A, Plotkin S. Set K-cover algorithms for energy efficient monitoring in wireless sensor networks. In: Proceedings of the 3rd international symposium on information processing in sensor networks, ACM, New York, pp. 424–432

[AMS06] Agha G, Meseguer J, Sen K (2006) PMaude: rewrite-based specification language for probabilistic object systems. Electron Notes Theor Comput Sci 153(2):213–239

[APM09] Audebaud P, Paulin-Mohring C (2009) Proofs of randomized algorithms in coq. Sci Comput Progr 74(8):568–589

[BMP08] Bernardeschi C, Masci P, Pfeifer H (2008) Early prototyping of wireless sensor network algorithms in PVS. In: Computer safety, reliability, and security. LNCS 5219. Springer, Berlin, pp 346–359

[BMP09] Bernardeschi C, Masci P, Pfeifer H (2009) Analysis of wireless sensor network protocols in dynamic scenarios. In: Stabilization, safety, and security of distributed systems, LNCS 5873. Springer, Berlin, pp 105–119

[Bog06] Bogachev VI (2006) Measure theory. Springer, Berlin

[CGP00] Clarke EM, Grumberg O, Peled DA (2000) Model checking. The MIT Press, Cambridge

[EHTA11] Elleuch M, Hasan O, Tahar S, Abid M (2011) Formal analysis of a scheduling algorithm for wireless sensor networks. In: Formal methods and software engineering, LNCS 6991. Springer, Berlin, pp 388–403

[EHTA13] Elleuch M, Hasan O, Tahar S, Abid M (2013) Formal probabilistic analysis of a wireless sensor network for forest fire detection. In: Symbolic computation in software science, EPTCS 122. Open Publishing Association, pp 1–9

[Ell13] Elleuch M (2013) Formalization of the detection properties of WSNs in HOL. HOL code. http://hvg.ece.concordia.ca/projects/prob-it/wsn.php

[Fel68] Feller W (1968) An introduction to probability theory and its applications, vol 1. Wiley, New York

[FHM07] Fehnker A, Van Hoesel L, Mader A (2007) Modelling and verification of the LMAC protocol for wireless sensor networks. In: Integrated formal methods, LNCS 4591. Springer, Berlin, pp 253–272

[Fru06] Fruth M (2006) Probabilistic model checking of contention resolution in the IEEE 802.15.4 low-rate wireless personal area network protocol. In: Proceedings of international symposium on leveraging applications of formal methods, verification and validation. IEEE Computer Society, New York, pp 290–297

[GM93] Gordon MJC, Melham TF (1993) Introduction to HOL: a theorem proving environment for higher-order logic. Cambridge Univ. Press, Cambridge

[HAA+09] Hasan O, Abbasi N, Akbarpour B, Tahar S, Akbarpour R (2009) Formal reasoning about expectation properties for continuous random variables. In: Formal methods, LNCS 5850. Springer, Berlin, pp 435–450

[Has08] Hasan O (2008) Formal probabilistic analysis using theorem proving. PhD thesis, Concordia Univ., Montreal

[Hew01] Hewish M (2001) Reformatting fighter tactics. Jane's Int Defense Rev. Jane's Information Group, London

[HH11] Hölzl J, Heller A (2011) Three chapters of measure theory in Isabelle/HOL. In: Interactive theorem proving, LNCS 6898. Springer, Berlin, pp 135–151

[HOL] The HOL theorem prover. http://hol.sourceforge.net/

[HRZ08] Hanna Y, Rajan H, Zhang W (2008) Slede: a domain-specific verification framework for sensor network security protocol implementations. In: Proceedings of conference on wireless network security. ACM, New York, pp 109–118

[HT07] Hasan O, Tahar S (2007) Formalization of continuous probability distributions. In: Automated deduction, LNCS 4603. Springer, Berlin, pp 3–18

[HT08] Hasan O, Tahar S (2008) Using theorem proving to verify expectation and variance for discrete random variables. Autom Reason 41(3–4):295–323

[Hur02] Hurd J (2002) Formal verification of probabilistic algorithms. PhD thesis, Univ. of Cambridge, Cambridge

[JS07] Jain S, Srivastava S (2007) A survey and classification of distributed scheduling algorithms for sensor networks. In: Proceedings of international conference on sensor technologies and applications. IEEE Computer Society, New York, pp 88–93

[LC08] Lin JW, Chen YT (2008) Improving the coverage of randomized scheduling in wireless sensor networks. IEEE Trans Wireless Commun 7(12):4807–4812

[Les07] Lester DR (2007) Topology in PVS: continuous mathematics with applications. In: Proceedings of the second workshop on automated formal methods. ACM, New York, pp 11–20

[Liu04] Liu C (2004) Randomized scheduling algorithm for wireless sensor neworks. In: Project report of randomized algorithm. University of Victoria, Victoria
[Liu13] Liu L (2013) Formalization of discrete-time markov chains in HOL. PhD thesis, Concordia Univ., Montreal, May 2013.
[LWXS06] Liu C, Wu K, Xiao Y, Sun B (2006) Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks. IEEE Trans Parallel Distrib Syst 17(6):562–575
[Mac98] MacKay DJC (1998) Introduction to Monte Carlo methods. In: Proceedings of NATO advanced study institute on learning in graphical models. Kluwer Academic Publishers, Dordrecht, pp 175–204
[Mha12] Mhamdi T (2012) Information-theoretic analysis using theorem proving. PhD thesis, Concordia Univ., Montreal, December 2012
[MHT10] Mhamdi T, Hasan O, Tahar S (2010) On the formalization of the lebesgue integration theory in HOL. In: Interactive theorem proving, LNCS 6172. Springer, Berlin, pp 387–402
[MHT11] Mhamdi T, Hasan O, Tahar S (2011) Formalization of entropy measures in HOL. In: Interactive theorem proving, LNCS 6898. Springer, Berlin, pp 233–248
[OT07] Ölveczky P, Thorvaldsen S (2007) Formal modeling and analysis of the OGDC wireless sensor network algorithm in real-time maude. In: Formal methods for open object-based distributed systems, LNCS 4468. Springer, Berlin, pp 122–140
[PRI] The PRISM model checker. http://www.prismmodelchecker.org/
[RKNP04] Rutten J, Kwaiatkowska M, Normal G, Parker D (2004) Mathematical techniques for analyzing concurrent and probabilisitc systems. In: CRM monograph series. American Mathematical Society, Providence
[RTM] The real-time tool. http://heim.ifi.uio.no/peterol/RealTimeMaude/.
[SWV+11] Sun Z, Wang P, Vuran MC, Al-Rodhaan AM, Al-Dhelaan AM, Akyildiz IF (2011) BorderSense: border patrol through advanced wireless sensor networks. Ad Hoc Netw 9(3):468–477
[XCW+10] Xiao Y, Chen H, Wu K, Sun B, Zhang Y, Sun X, Liu C (2010) Coverage and detection of a randomized scheduling algorithm in wireless sensor networks. IEEE Trans Comput 59(4):507–521
[XZP+09] Xiao Y, Zhang Y, Peng M, Chen H, Du X, Sun B, Wu K (2009) Two and three-dimensional intrusion object detection under randomized scheduling algorithms in sensor networks. Comput Netw 53(14):2458–2475
[XZSC07] Xiao Y, Zhang Y, Sun X, Chen H (2007) Asymptotic coverage and detection in randomized scheduling algorithm in wireless sensor networks. In: Proceedings of international conference on communications. IEEE, New York, pp 3541–3545
[YMG08] Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330
[ZBA10] Zayani H, Barkaoui K, Ben Ayed R (2010) Probabilistic verification and evaluation of backoff procedure of the WSN ECo-MAC protocol. Int J Wirel Mobile Netw 2(2):156–170
[ZSL+11] Zheng M, Sun J, Liu Y, Dong JS, Gu Y (2011) Towards a model checker for NesC and wireless sensor networks. In: Formal methods and software engineering, LNCS 6991. Springer, Berlin, pp 372–387

*Received 7 January 2014*
*Revised 4 April 2014*
*Accepted 5 June 2014 by Jin Song Dong*
*Published online 12 July 2014*